Data Science & Statistics

Tutorial-1

Probability Addition Theorem:

Definition: The addition theorem of probability states that the probability of the union of two events is equal to the sum of their individual probabilities minus the probability of their intersection.

Formula: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

Example:
What is the probability of drawing a king or a heart from a standard deck of 52 cards?

Solution:

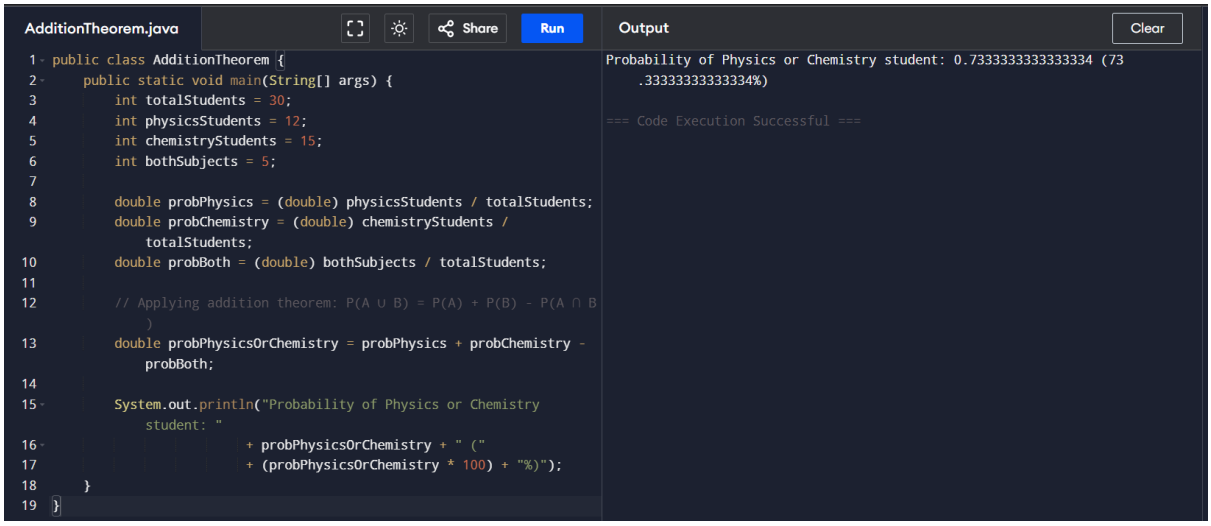
- P(King) = 4/52
- P(Heart) = 13/52
- P(King ∩ Heart) = 1/52
- P(King ∪ Heart) = 4/52 + 13/52 - 1/52 = 16/52 ≈ 0.3077

Java Code:

```
public class AdditionTheorem {
   public static void main(String[] args) {
      int totalStudents = 30;
      int physicsStudents = 12;
      int chemistryStudents = 15;
      int bothSubjects = 5;

      double probPhysics = (double) physicsStudents / totalStudents;
      double probChemistry = (double) chemistryStudents / totalStudents;
      double probBoth = (double) bothSubjects / totalStudents;

      // Applying addition theorem: P(A ∪ B) = P(A) + P(B) - P(A ∩ B)
      double probPhysicsOrChemistry = probPhysics + probChemistry - probBoth;
```

```
        System.out.println("Probability of Physics or Chemistry student: "
                + probPhysicsOrChemistry + " ("
                + (probPhysicsOrChemistry * 100) + "%)");
    }
}
```

Execution:



Probability Multiplication Theorem:

Definition: The multiplication theorem states that the probability of two events occurring together (their intersection) is equal to the probability of one event multiplied by the conditional probability of the other event given the first has occurred.

Formula: P(A ∩ B) = P(A) * P(B|A)

Example:
A bag contains 5 red balls and 4 blue balls. Two balls are drawn successively without replacement. What is the probability that both balls are red?

Solution:

- P(First red) = 5/9
- After drawing one red ball, P(Second red | First red) = 4/8

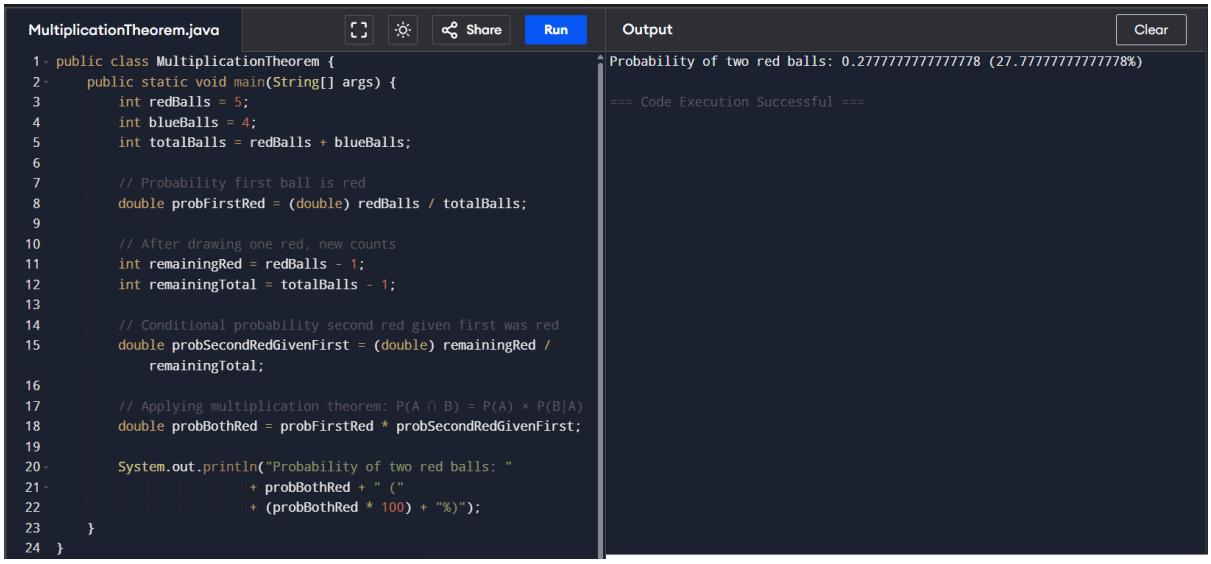
Using multiplication theorem:
P(Both red) = P(First red) × P(Second red | First red)
= (5/9) × (4/8)
= (5/9) × (1/2)
= 5/18 ≈ 0.2778 or 27.78%

Java Code:

```
public class MultiplicationTheorem {
  public static void main(String[] args) {
    int redBalls = 5;
    int blueBalls = 4;
    int totalBalls = redBalls + blueBalls;

    // Probability first ball is red
    double probFirstRed = (double) redBalls / totalBalls;

    // After drawing one red, new counts
    int remainingRed = redBalls - 1;
    int remainingTotal = totalBalls - 1;

    // Conditional probability second red given first was red
```

```
        double probSecondRedGivenFirst = (double) remainingRed / remainingTotal;

        // Applying multiplication theorem: P(A ∩ B) = P(A) × P(B|A)
        double probBothRed = probFirstRed * probSecondRedGivenFirst;

        System.out.println("Probability of two red balls: "
                + probBothRed + " ("
                + (probBothRed * 100) + "%)");
    }
}
```

Execution:



Bayes' Theorem:

Definition: Bayes' theorem describes the probability of an event based on prior knowledge of conditions that might be related to the event. It relates the conditional and marginal probabilities of random events.

Formula: P(A|B) = [P(B|A) * P(A)] / P(B)

Example:
A factory has two machines producing widgets:

- Machine A produces 60% of widgets, with 2% defective rate
- Machine B produces 40% of widgets, with 3% defective rate

If a randomly selected widget is defective, what is the probability it came from Machine A?

Solution:

Let:

- P(A) = 0.6 (probability from Machine A)
- P(B) = 0.4 (probability from Machine B)
- P(Defective|A) = 0.02
- P(Defective|B) = 0.03

First find total probability of defect:
P(Defective) = P(Defective|A)P(A) + P(Defective|B)P(B)
= (0.02 × 0.6) + (0.03 × 0.4)
= 0.012 + 0.012
= 0.024

Now apply Bayes' Theorem:
P(A|Defective) = [P(Defective|A) × P(A)] / P(Defective)
= (0.02 × 0.6) / 0.024
= 0.012 / 0.024
= 0.5 or 50%

Java Code:

```
public class BayesTheorem {

  public static void main(String[] args) {

    // Machine A produces 60% of widgets with 2% defective

    double probMachineA = 0.6;
```

```
        double defectRateA = 0.02;

        // Machine B produces 40% of widgets with 3% defective
        double probMachineB = 0.4;
        double defectRateB = 0.03;

        // Total probability of defect (denominator in Bayes' formula)
        double totalProbDefect = (defectRateA * probMachineA)
                        + (defectRateB * probMachineB);

        // Applying Bayes' theorem: P(A|Defect) = [P(Defect|A) × P(A)] / P(Defect)
        double probAGivenDefect = (defectRateA * probMachineA) / totalProbDefect;

        System.out.println("Probability defective came from Machine A: "
                + probAGivenDefect + " ("
                + (probAGivenDefect * 100) + "%)");

        // Bonus: Calculate for Machine B as well
        double probBGivenDefect = (defectRateB * probMachineB) / totalProbDefect;
        System.out.println("Probability defective came from Machine B: "
                + probBGivenDefect + " ("
                + (probBGivenDefect * 100) + "%)");
    }
}
```

Execution:

BayesTheorem.java

Share Run

```java
public class BayesTheorem {
    public static void main(String[] args) {
        // Machine A produces 60% of widgets with 2% defective
        double probMachineA = 0.6;
        double defectRateA = 0.02;

        // Machine B produces 40% of widgets with 3% defective
        double probMachineB = 0.4;
        double defectRateB = 0.03;

        // Total probability of defect (denominator in Bayes' formula)
        double totalProbDefect = (defectRateA * probMachineA)
                               + (defectRateB * probMachineB);

        // Applying Bayes' theorem: P(A|Defect) = [P(Defect|A) × P(A)] / P(Defect)
        double probAGivenDefect = (defectRateA * probMachineA) / totalProbDefect;

        System.out.println("Probability defective came from Machine A: "
                           + probAGivenDefect + " ("
                           + (probAGivenDefect * 100) + "%)");

        // Bonus: Calculate for Machine B as well
        double probBGivenDefect = (defectRateB * probMachineB) / totalProbDefect;
        System.out.println("Probability defective came from Machine B: "
                           + probBGivenDefect + " ("
                           + (probBGivenDefect * 100) + "%)");
    }
}
```

Output

```
Probability defective came from Machine A: 0.5 (50.0%)
Probability defective came from Machine B: 0.5 (50.0%)

=== Code Execution Successful ===
```